

A Multi-Operator Simulation for Investigation of Distributed Air Traffic Management Concepts

Mark E. Peters,^{*} Mark G. Ballin,[†] John S. Sakosky[‡]
Hampton, VA 23666

Abstract

This paper discusses the current development of an air traffic operations simulation that supports feasibility research for advanced air traffic management concepts. The Air Traffic Operations Simulation (ATOS) supports the research of future concepts that provide a much greater role for the flight crew in traffic management decision-making. ATOS provides representations of the future communications, navigation, and surveillance (CNS) infrastructure, a future flight deck systems architecture, and advanced crew interfaces. ATOS also provides a platform for the development of advanced flight guidance and decision support systems that may be required for autonomous operations.

Introduction

The Air Traffic Operations Simulation (ATOS) is a mid-fidelity simulation under development to support research of air traffic operations within future airspace environments. Hosted by the Air Traffic Operations Laboratory at the NASA Langley Research Center, ATOS is a workstation-based human-in-the-loop simulation that serves as a test bed for investigations of future distributed air/ground traffic management concepts and their associated decision support tools. This paper describes the ATOS capability, its design, and its application to advanced traffic management research. The paper first presents a brief background of the future concepts under study and describes the simulation capabilities

needed for their development and assessment. An overview of the simulation architecture is then provided with explanations of the design choices that were made. Elements of the future airspace system represented in the initial build of ATOS are then described. These elements include the aircraft components and their future avionics capabilities, the airborne and ground-based communications, navigation, and surveillance / air traffic management (CNS/ATM) infrastructure, and future system services.

Background

The Distributed Air Ground Traffic Management (DAG-TM) concept, proposed by NASA, describes a set of future traffic operations that redefines the roles of flight crews, air traffic service providers, and aeronautical operational control organizations.¹ It is based on the premise that the sharing of information between these system participants and the delegation of decision authority to the most appropriate decision maker will result in large improvements to airspace system capacity and robustness.² Distributed decision making authority may also be a key to enabling large increases in airspace user efficiency and flexibility, and in minimizing human workload bottlenecks that limit system capacity.

DAG-TM builds upon the concept of “revolutionary evolution” advocated by the Future Air Navigation System (FANS), which cleared the way for a distributed air-, ground-, and space-based CNS/ATM infrastructure. DAG-TM also utilizes the idea of user-optimal routing efficiencies as advanced by the Free Flight paradigm, in which airspace users are free to select their path and speed in real-time. To do this, advanced airborne systems are needed to enable autonomous airborne flight planning in the presence of traffic, terrain, and weather hazards. These systems build upon the concepts of Traffic Collision Avoidance System (TCAS), Terrain Awareness and Warning System

Copyright © 2002 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Government purposes. All other rights are reserved by the copyright owner.

^{*} Systems Engineer, Seagull Technology, Member, AIAA

[†] Research Engineer, NASA Langley Research Center, Associate Fellow, AIAA

[‡] Senior Systems Engineer, SAIC, Senior Member, AIAA

(TAWS), and Windshear/Weather radar. Through secure multi-layered data management supported by systems such as Active Dependent Surveillance-Broadcast (ADS-B), Controller Pilot Data Link Communications (CPDLC), Flight Information Service (FIS), and Traffic Information Service (TIS), an increased awareness of hazards and constraints should allow crews to have greater authority and responsibility in managing their flights.

The simulation of airspace and the operations of aircraft within it is a useful and common approach for the study of new air traffic management concepts, decision support automation, and procedures. For instance, the Pseudo Aircraft System was developed to facilitate the development of the Center/TRACON Automation System.³ Additionally, the Federal Aviation Administration (FAA) uses its Target Generation Facility to study operational air traffic control problems,⁴ which has recently been enhanced to include a higher fidelity dynamics model.⁵ For the study of future concepts such as DAG-TM, a traffic operations simulation is needed to study the interactions of distributed human decision makers, especially regarding the transfer of responsibility between airborne and ground-based human operators. It also facilitates the study of supporting systems, such as decision support tools and future CNS infrastructure, and their impacts on concept feasibility. As feasibility requirements and bounds are established, traffic operations simulation is needed to support detailed design and evaluation of air and ground-based decision support automation, computer/human interfaces, and operational procedures. When these activities are sufficiently mature, traffic operations simulation can facilitate the assessment of concept benefits and economic viability.

Simulation Requirements for Research

There are 6 main requirements that a simulation must meet to be useful as a tool for (CNS/ATM) research and development:

Multiple Human Operators

The simulation must support multiple human operators, each of which is provided a level of sophistication and fidelity that enables them to make traffic management decisions as they would in the actual environment.

System Flexibility with Variable Fidelity

The simulation must be flexible enough to accommodate the different levels of fidelity and different levels of equipage. Workstation simulations are usually adequate to simulate multi-aircraft scenarios with human operators in an affordable and manageable way, however, high-fidelity simulations are occasionally desired to duplicate a full-crew and full-workload environment, and to aid in the development of procedures. The CNS/ATM infrastructure representation must also be flexible to compare future CNS concepts with today's system. Since the future CNS infrastructure will also be required to accommodate many legacy systems, a range of present and future CNS components and aircraft equipage levels must be accommodated simultaneously.

Simulation Repeatability

For proper experimental control, the simulation should produce repeatable results. Test subjects must be provided with identical situations, so that only test subject actions produce differing results. Additionally, some experiments will test algorithms, such as traffic conflict resolution, which can produce large response differences to small input differences. Simulation repeatability for functions such as these is very important for the verification and validation of algorithms.

Modes Of Operation

The simulation must employ two modes of operation. These are: human in the loop (HITL) and human operator modeled batch. HITL is defined as a mode in which subject pilots, subject controllers, and/or subject dispatchers interact with systems or simulators that allow them to make flight-management or traffic-management decisions as they would in a deployed system. Human operator modeled batch is a mode in which all system operations are simulated with full repeatability, using automated human operator models to represent humans in the simulated system.

Component Error Modeling

Success of DAG-TM and other future concepts hinges upon the quality of information available to the decision maker. Required Navigation Performance (RNP) compares actual aircraft position against required position. Required Communication Performance (RCP) states operational performance in terms of communication process time, integrity,

availability, and continuity of function. Required Surveillance Performance (RSP), yet to be defined, will standardize requirements for surveillance accuracy. Parametric control of RNP, RCP, and RSP levels as well as the accuracy of system predictions is needed to understand the impacts of the associated errors on concept feasibility. The simulation must also independently generate and track information that represents actual (truth) values from that which represents values available to simulation components and participants.

Simulation Execution Control

Simulation execution control is normally used in flight simulators that utilize a real-time simulation environment. To be compatible with such simulations, several simulation mode control functions must be accommodated. These are: initialize, trim, hold, operate, and reset.

Simulation Evolution

The development of ATOS, formerly known as Free Flight Simulation,⁵ was initiated after an investigation of existing traffic management simulations revealed that none met the above goals. Existing real-time simulations were not flexible enough to model future as well as current operational environments and they lacked the mode control needed for rigorous and formal research. Existing batch assessment tools were not suitable because they do not represent human decision-making. They also lack the detailed flight deck and service provider representations required for design of decision support automation tools.

Initial goals for the simulation also included an aggressive development schedule that would provide research support as soon as possible. Toward this goal, a prototype simulation was developed that made much use of existing capabilities. It allows several workstation-based, real-time aircraft simulators to interact with each other and with target aircraft generated by a central airspace simulation. Existing software codes developed by NASA and National Aerospace Laboratory of the Netherlands (NLR) were modified to create the initial prototype,^{6, 7} which has been used to investigate the use of traffic intent information in constrained air-traffic operations⁸.

A new version of the simulation is under development to provide higher fidelity representations of system elements and improved utility to researchers and technology developers. Referred to as ATOS Build 1, it resolves several limitations of the prototype simulation. Among the limitations was the reliance on target generators for all aircraft other than the test subject stations. Target generators were developed to enable many simulated aircraft to be controlled by pilot operators (referred to as “pseudo-pilots”), who in turn receive voice clearances and instructions from ground-based human controllers. However, the aircraft represented by these simulations normally have no capability to avoid conflicts, and it is impractical to expand their capability. These simulations, while appropriate to support ground-based air traffic control research, do not adequately support DAG-TM research requirements for large numbers of aircraft and their systems that enable autonomous operations.

Design Overview

In addition to the identified research requirements, which focus on the research utility of the simulation, several software design goals were set to facilitate scalability and ease of maintenance:

- Maximize simulation scalability, so that large numbers of aircraft can be simulated without exceeding computational speed limitations of any process
- Develop a simulation architecture that avoids the development and maintenance of duplicate functions
- Include a capability to duplicate and distribute simulation components among several facilities, via dedicated connections and possibly via the internet
- Require few or no changes to flight deck decision support technology to interface with ATOS, Langley high-fidelity simulations, and Langley research aircraft such as the NASA Airborne Research Integrated Experiments System (ARIES), a specially instrumented Boeing 757 that is used to support in-flight validation activities.

ATOS Build 1 is an extensible distributed system consisting of many independent processes that operate and communicate with each other through the network. The Build 1 system components are illustrated in Figure 1. The system architecture is flexible to allow for experimental investigations of competing systems developed for autonomous operations.

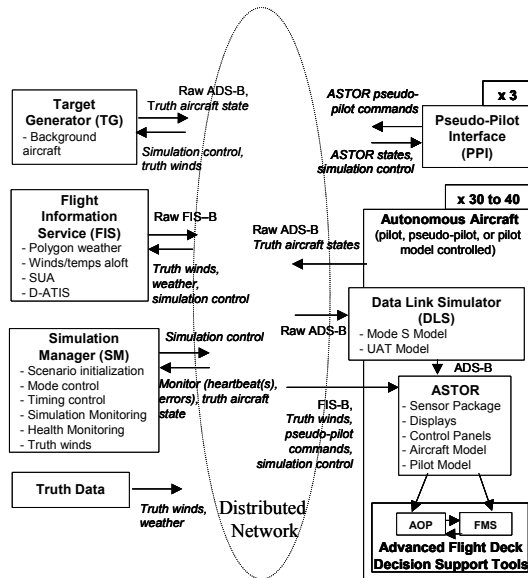


Figure 1. Illustration of the distributed system, which enables flexibility and extensibility

An independent aircraft simulation, referred to as the Aircraft Simulation for Traffic Operations Research (ASTOR), is employed for each aircraft represented within ATOS Build 1. These workstation-based, HITL aircraft simulations have 6 degree of freedom dynamic models supported by actual aircraft aerodynamic data. The dynamics model can simulate jet and piston aircraft, and the simulation is equipped with representative cockpit displays and equipage levels for the different types of aircraft. Furthermore, the models are extensible to different aircraft types by simple aerodynamic/propulsion parameter changes.

The aircraft simulations can be executed with pilot test subjects in HITL operation, or pseudo-pilots can drive them as background traffic. This form of background traffic is much more capable than target generated traffic and can be expected to have the same behavior as the HITL aircraft. If necessary, additional traffic can still be provided by target generation software for large simulations, since the currently existing target

generation capability is not compromised. With the addition of a pilot model for each aircraft simulation, background traffic can be represented without the need for pseudo-pilots, or the entire simulation can be executed in a batch mode to support Monte-Carlo analysis.

ASTOR also models a generic future flight deck architecture that contains air/air and air/ground datalinks, and advanced crew interfaces, and it hosts the flight management systems and decision support systems required for autonomous operations. These processes are independent from the aircraft simulation code itself, to provide accurate representations of aircraft equipage. The simultaneous use of various versions of ASTOR enables the investigation of operations involving various aircraft types with differing equipage levels as well as widely varying performance and operating characteristics.

Simulation Architecture

To support such generalized simulations involving a large number of independent systems, a sophisticated software network is required.

HLA (High Level Architecture)

The ATOS infrastructure is based on the Department of Defense's (DoD) High Level Architecture (HLA)⁹. The use of HLA to implement the interfaces between the ATOS components enables a widely distributed simulation limited only by the available hardware resources on the network. Each ATOS component, known as a Federate in the HLA paradigm, communicates through the distributed network, eliminating the need for centralized control of data flow that has in the past limited the extensibility of the simulation. Furthermore, the use of HLA enables the inclusion of other simulation assets such as full cockpit simulations that require remote access.

Federation Components

Each ATOS component is a separate software process that usually operates on hardware dedicated for that process. These processes include aircraft simulations, airport models, and target generators. The simulation manager is a federate that is used to manage the simulation. The only non-federate is the Run Time

Infrastructure (RTI) exec, an HLA process which manages HLA communications and other HLA specific tasks. Figure 2 illustrates the components of the Federation.

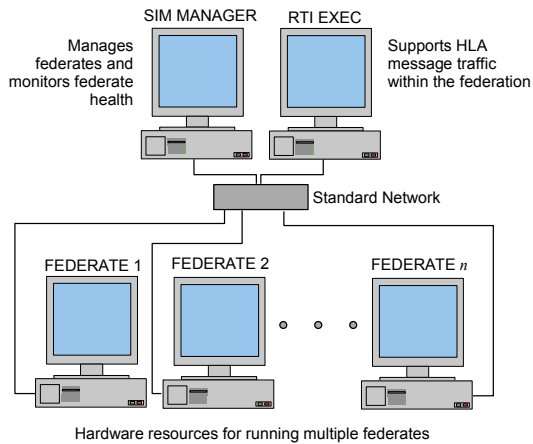


Figure 2. Illustration of a Federation, the simulation architecture when using HLA

RTI Executive and Interactions

The Real Time Interface (RTI) executive, which is part of HLA, monitors the joining of the federation by federates and manages the various messages that can be sent within the federation. Within the HLA paradigm, these messages are referred to as interactions. The number and types of interactions are defined at the beginning of the simulation by the RTI exec and its associated data files. Each federate subscribes to a subset of the total available interactions when it is initialized. The federate also agrees to publish certain types of interactions. Examples of these interactions include aircraft state information and simulation state modes. The end result is that the any federate can send or receive information to/from any other federate directly.

Simulation Manager

The simulation manager is the primary simulation control and monitoring station. The simulation control window, illustrated in Figure 3, controls the simulation state and monitors the health status of all the federates during the simulation. It can command the federates into the various states such as reset, hold, and operate. The second window, the scenario and event window, can record important events, collect data, and send specialized instructions to a

particular federate through its real-time scripting capability. The scenario and

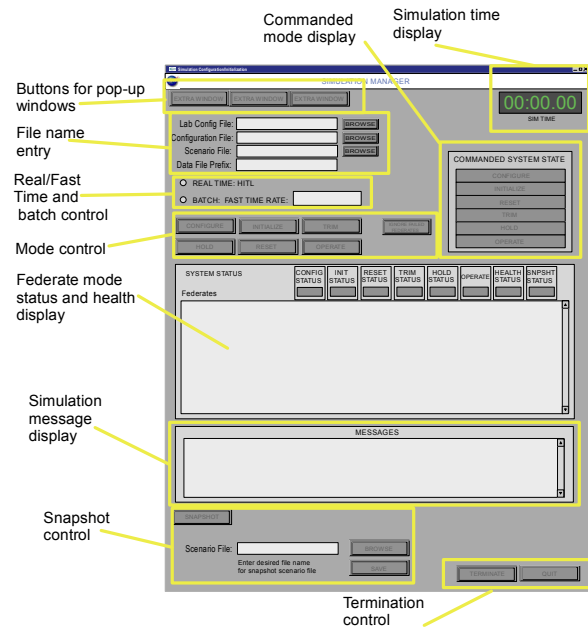


Figure 3. Detailed explanation of the simulation control window

event window is pictured in Figure 4. The scenario manager also has a planview display which allows the operator to monitor the airspace being modeled.

Simulation Modes

The simulation manager controls transition to and from various system modes. Figure 5 shows the simulation modes and the possible transitions between them. The modes include: Configure, Reset, Trim, Hold, Operate, and Terminate.

Configure

The CONFIGURE mode is the mode that the federate application enters after being started. The CONFIGURE mode cannot be transitioned to from any other simulation modes. Configuring

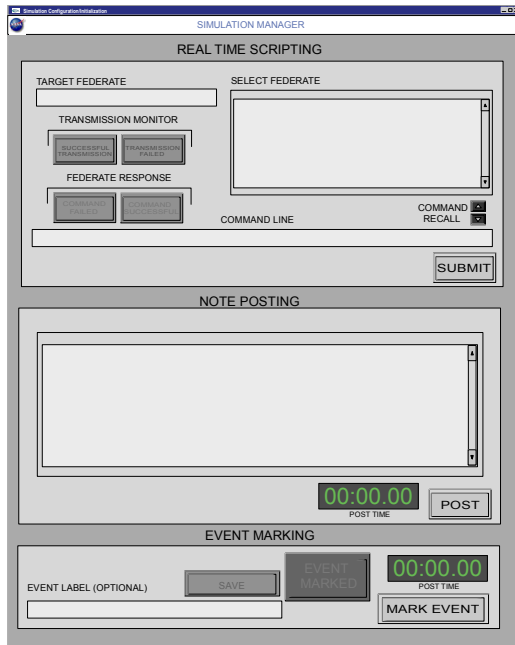


Figure 4. Illustration of the Scenario and Event window

consists of initializing federates and all non-scenario related data.

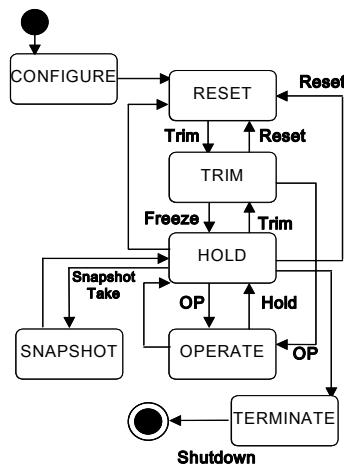


Figure 5. Simulation Modes and Transitions

Reset

The RESET mode applies or reapplies initial conditions.

Trim

The TRIM mode allows federates to reach a steady state before running the current scenario. For aircraft federates, the TRIM mode may involve determining the control surface deflections and power settings required to reach equilibrium. For other federates that do not need to achieve a steady-state condition, TRIM may simply be a “do nothing” mode.

Hold

The HOLD mode is one where the federates have been initialized and trimmed and is ready to run the scenario. The HOLD mode also occurs when an active simulation has been temporarily suspended. In the HOLD mode, the simulation clock does not advance, however, user interface displays are allowed to change

Operate

The OPERATE mode is where the federates are running the simulation scenario.

Snapshot

The SNAPSHOT feature saves the current state of the system to disk so that it can be used as the starting point of a future scenario. Each component is required to identify and save its own mode information needed to meet this objective.

Terminate

The TERMINATE mode shuts down the federates. Once the component has exited, it will need to be re-launched in order to be used again.

Health Monitoring

The simulation manager maintains a record of the system health. To do this, each federate sends a periodic message indicating its current state and its current health status. There are three health statuses: Normal, Warning and Failed. The federate status is presented to the operator on the simulation control window in green, yellow, or red indicator lights. If a federate completely fails, and is unable to send any messages back to the simulation manager, the simulation manager flags the federate as failed and posts a red indication after a specified number of missed messages.

Figure 6 illustrates a failure of one of the federates. In this case, the overall system status

row indicates a failed federate somewhere within the simulation. The operator is then alerted to determine which federate has failed by scrolling through the federate list, or by reading failure messages within the message box.

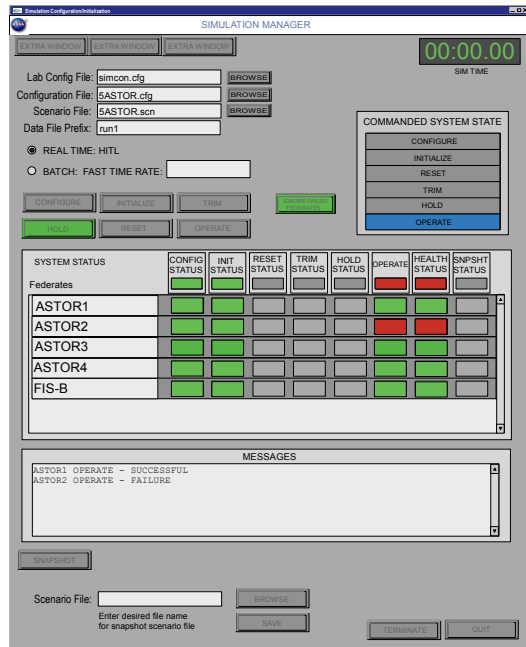


Figure 6. Illustration of the simulation manager indicating a failure on ASTOR 2

Real Time Scripting

Under certain conditions, the operator may wish to send a specific set of instructions to a federate while the system is running. Examples might include sending a new route to an aircraft simulation federate, or a different data file name to a FIS federate. This task is accomplished using the top portion of the scenario and event window as illustrated in Figure 7. The operator can choose a federate to receive the message from the federate list. Then a string is typed in the message box. When the string is sent to the appropriate federate, the simulation manager responds with two confirmations. These are: 1. The message was received, and 2. The message was understood and complied with.

Event Marking

The scenario and event marker also enables the operator to mark interesting or noteworthy events. The operator marks the event and then

types a short message indicating the significance of the event. This feature enables the researcher to easily find the spot during data analysis.

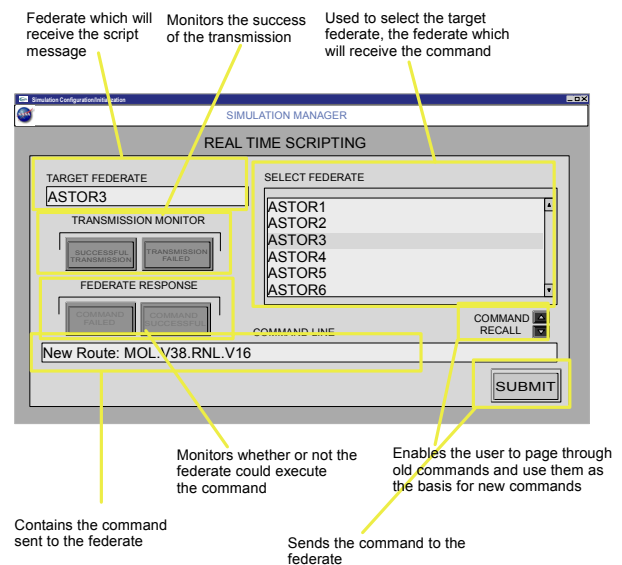


Figure 7. Illustration of the real time scripting feature in the scenario and event window

Note Posting

The operator can also take notes with a small text editor that is positioned in the middle of the Scenario and Event window. These notes are posted to the data file when they are submitted and do not pertain to any particular event.

Data Collection

Each federate is responsible for collecting its own data. The simulation manager is responsible for instructing the federates as to whether data is to be collected. The simulation manager collects its own data which consists of all messages received from the federates, all notes, and posted events.

Operating Modes

The ATOS software supports an emulated real-time and a batch mode of operation.

Emulated Real Time

The emulated real-time mode is equivalent to real-time (or synchronous) operation, except that a real-time operating system is not used. In emulated real-time mode, a frame overrun is not an error. Components can attempt to “catch up” by immediately re-executing the task instead of

waiting for the next clock tick. The emulated real-time operating mode has two sub-modes: human in the loop (HITL), as illustrated in Figure 8, and fast time. In HITL mode, the simulation time always advances at a normal real-time rate, i.e. at the same rate as a wall clock. The fast-time mode is implemented as the “speed times N function” defined in ARINC 610A¹⁰.

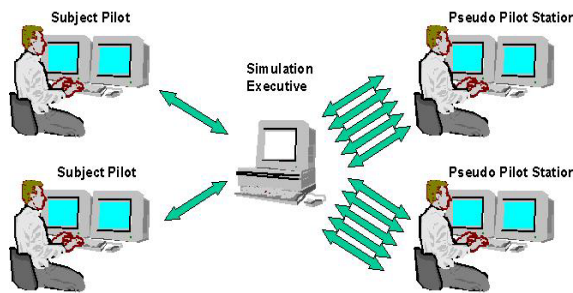


Figure 8. Illustration of HITL mode consisting of subject pilot workstations with high fidelity background traffic provided by pseudopilots

Batch Mode

In batch mode, the executions of periodic tasks are not synchronized to a clock. Periodic tasks run at the fastest rate that the underlying hardware and operating system allows, or the clock tick is substituted for an asynchronous event. Unlike emulated real-time operation, event-driven tasks may be allowed to run without preemption by the periodic tasks. Batch mode is illustrated in Figure 9.

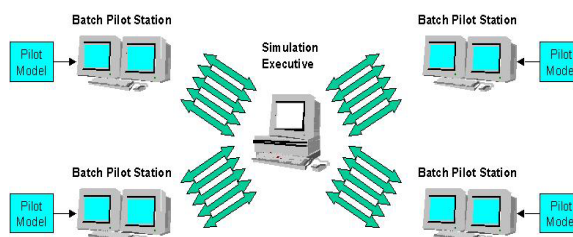


Figure 9. Illustration of batch mode functionality

While the simulation architecture supports batch and fast time, not all federates may be able to do this. The use of these tools is dependent on the nature of the individual federates that comprise the simulation.

Time Synchronization

The simulation manager controls the master clock of the simulation and periodically sends out a time-stamped message to all of the federates. This message is used by the federates to insure they are synchronized with the master clock.

Aircraft Simulations

While any aircraft simulation that complies with the basic federate infrastructure of ATOS can be used, the primary aircraft simulation is ASTOR. ASTOR supports four objectives:

- Advanced mid-fidelity aircraft simulation with enough sophistication to study DAG-TM concept feasibility while remaining simple enough to execute on a workstation platform.
- Modular architecture, which facilitates simulation of various aircraft types, each with unique airframe/engine and autopilot/autothrottle models, flight deck displays/controls, and various avionics equipage levels.
- Advanced avionics modeling representative of the future CNS/ATM environment.
- Capable of operating in several configurations to support research requiring piloted aircraft, pseudo-piloted aircraft, and aircraft with modeled pilots (batch mode).

Mid-Fidelity Workstation Based Simulation

ASTOR was conceived as an instance of the Langley Standard Real-Time Simulation (LaSRS) framework¹¹. LaSRS has successfully been applied to a spectrum of aircraft simulations ranging from mid-fidelity workstation-based general aviation simulations to high-fidelity full-motion cockpit air transport category simulations. As such the flexibility of this framework is proven.

The DAG-TM concept may rely on advanced trajectory management functionality provided by the autonomous operations planner (AOP),¹² which continuously provides a conflict free route to the crew, and real time flight management (FMS) subsystems. The FMS guidance function may need to provide continuous closed loop (temporal and spatial, i.e., 4D) guidance commands to the auto-flight /auto-throttle systems. ASTOR must host these advanced functions for each simulated aircraft that is equipped for autonomous operations.

Matched performance across the aero / mass / propulsion, auto-flight / auto-throttle and AOP/FMS models is essential for viable DAG-TM research. This need for matched and appropriate levels of representation for traffic management research in turn drives the need for the accurate modeling of climb, cruise, idle-thrust descent, and final approach phases of flight; dynamic aircraft mass as a function of fuel burn, which affects climb and descent performance; and throttle/thrust lever position, altitude, and Mach effects on engine dynamics.

Modular Architecture

To meet the meta-simulation requirements associated with DAG-TM research and airborne technology development, all aircraft type-specific features such as aerodynamic data, mass

properties, propulsion, flight controls, displays, and avionics have been abstracted from the

generic six degree of freedom ASTOR simulation framework. Models for a large array of aircraft types with widely varying performance and equipage levels are in development.

Figure 10 through Figure 12 show the ASTOR Flight Deck displays, which are representations of currently available equipment enhanced for autonomous operations.

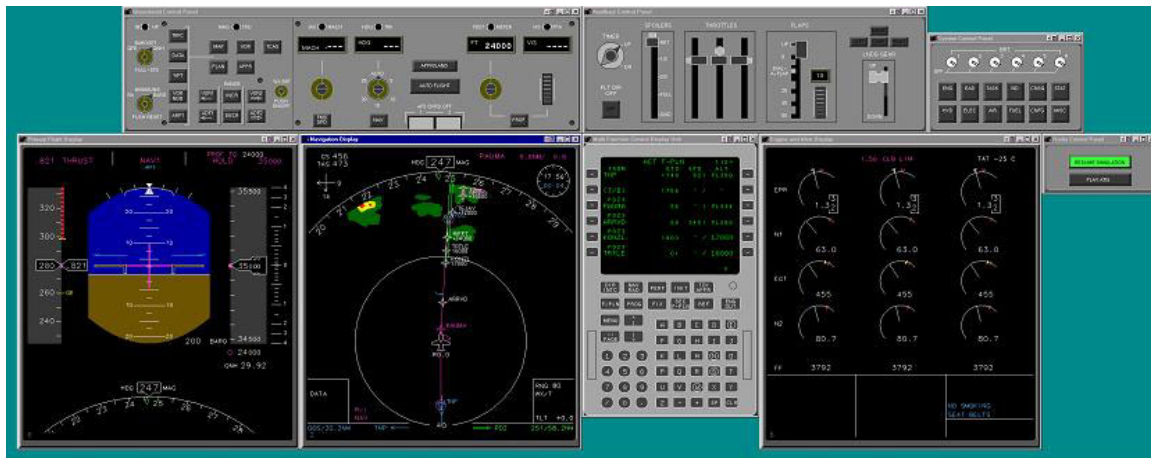


Figure 10. MD11 ASTOR Pilot Station



Figure 11. Boeing 777 ASTOR Pilot Station

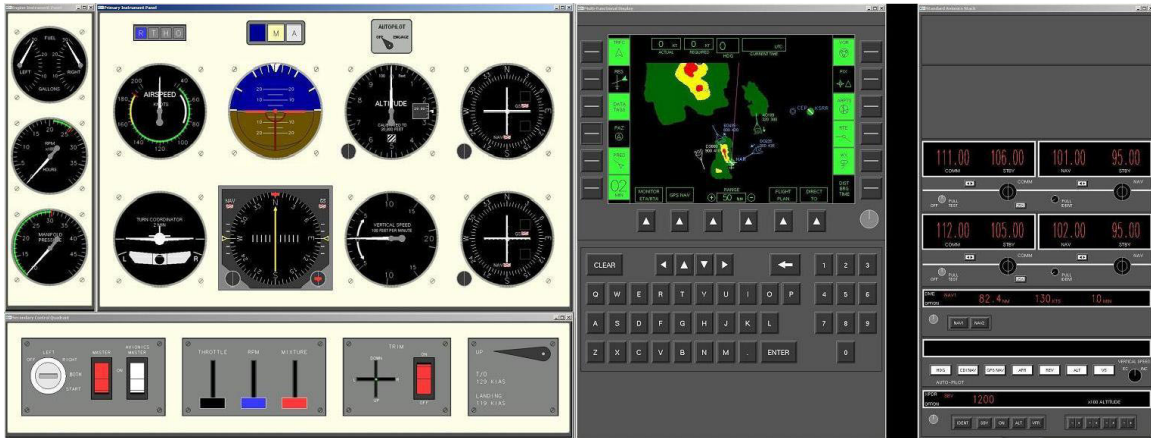


Figure 12. Lancair C300 ASTOR Pilot Station

Advanced Avionics Modeling

The ASTOR design focuses on a modular avionics architecture which achieves a high level of functional integration in the spirit of state of the art integrated modular architectures for avionics as employed in the commercial Boeing 777 and military F22 designs.¹³

This additional level of fidelity supports a fundamental research goal to enable ASTOR to not only assist in concept level development, but also provide insight in to the design of the actual hardware that would be required in a real flight deck.

ASTOR bridges the gap between system-level design / assessment tools and detailed subsystem designs, which are needed to support implementation of future concepts, but often can not easily be tested in a distributed environment. ASTOR provides an appropriate level of modeling in terms of information transfer, and information quality (RNP, RCP, RSP). These attributes enable the prototyping of various flight deck concepts with a level of operational realism that can address feasibility, robustness, and certification concerns as necessary for the validation of the resultant research data. This type of interaction between subsystems is critical to understanding the detailed flight deck design requirements.

The flexible nature of the avionics architecture enables ASTOR to represent a spectrum of actual aircraft equipage levels. Fundamental design components (e.g. GPS, IRS, FCC, TMC, and FMS) can be individually selected for a particular equipage level. These interfaces between components are also explicitly modeled and represent realistic avionics architectures at a high level. The flexibility of ASTOR enables researchers to consider/understand/model both legacy systems and industry modernization plans such as ARINC660A.

Integrated CNS Avionics Architecture

The advanced autonomous flight deck features embodied in the AOP and FMS subsystems are integrated into a CNS/ATM avionics architecture compatible with ARINC660A. The ARINC 660A CNS/ATM standard provides key design guidance for CNS functional partitioning. The ARINC 429 DITS data bus is simulated for the widest range of reverse and forward compatibility.

Figure 13 illustrates the integration of AOP and FMS elements of the Future Autonomous Flight Deck into an ARINC 660A CNS/ATM avionics architecture.

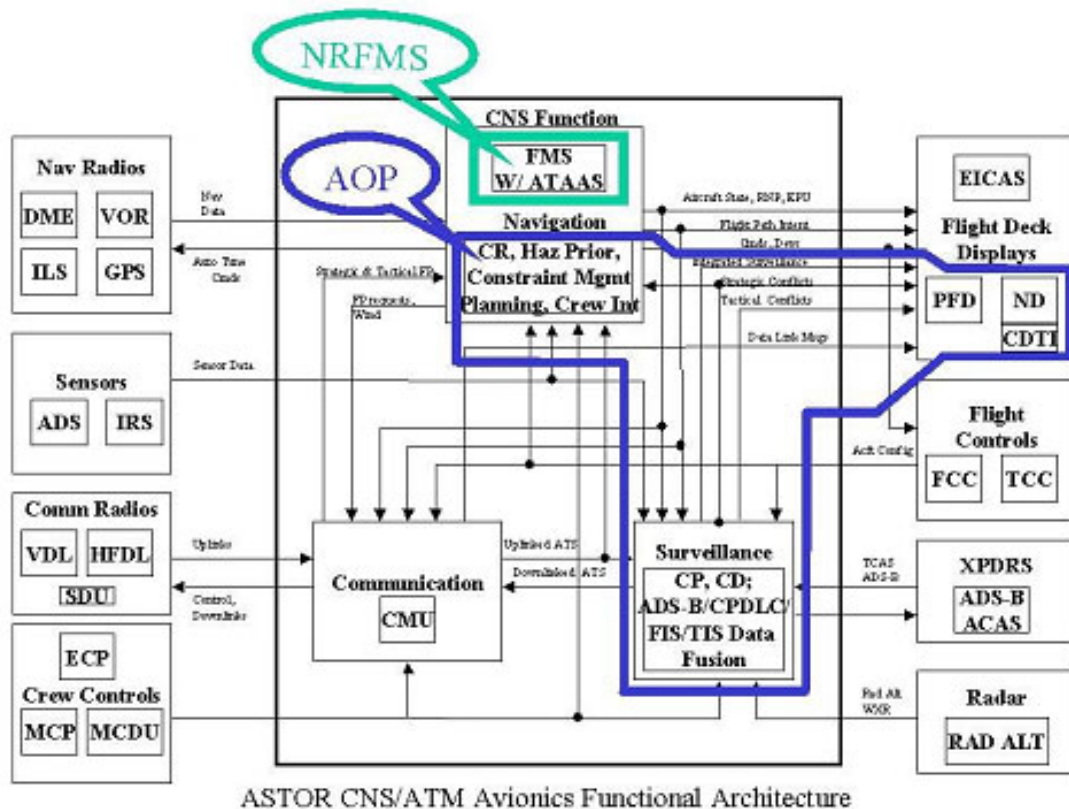


Figure 13. ASTOR Architecture of the Future Autonomous Flight Deck

Avionics Bus Fault Modeling

Full-featured error and failure modes such as total failure, signal bias failure, intermittent signal failure, and random noise failure are modeled within the Avionics Bus package.

ADS-B Datalink Simulations

The ADS-B subsystem allows the transmission of on board data to air or ground based users via a data link (e.g. Mode S, VDL-4, UAT) using a broadcast mode. ADS-B equipped aircraft and vehicles automatically broadcast important information -- latitude and longitude, velocity, altitude, heading, identification and, optionally, intent -- as determined by the avionics on board. The ATOS simulation provides several datalink simulation (DLS) options. These simulations are based on a mathematical datalink model and

implemented using a distributed software architecture.

Centralized and Distributed Solutions

There are trade-offs between a centralized DLS model and a distributed DLS model. The DLS architecture can be either centralized, where a single process collects all messages and then sends a subset of the messages to each federate aircraft, or it can be distributed, where each federate must simulate the DLS for itself. Figure 14 and Figure 15 illustrate the centralized and distributed solutions respectively.

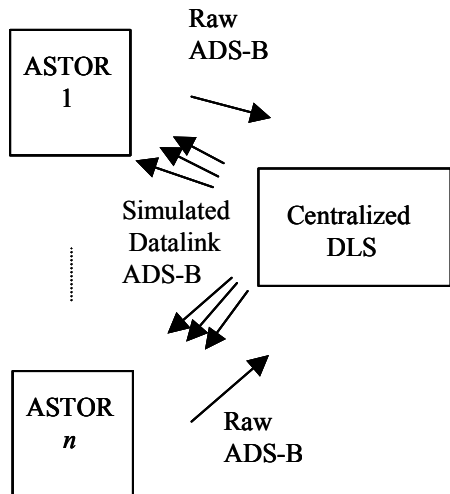


Figure 14. Illustration of a centralized DLS

Computational Comparison

When simulating the DLS in a centralized architecture, all messages must be gathered at a central location and then rebroadcast according to the algorithms within DLS model. The centralized approach is simple because only one process is needed, but it can become a simulation bottleneck that requires very high computational speed if the number of aircraft becomes large. The design limits the scalability of the system.

The distributed approach does not reduce the amount of computation required, but rather it distributes the computational load to all the aircraft. This eliminates a computational bottleneck on a single process but

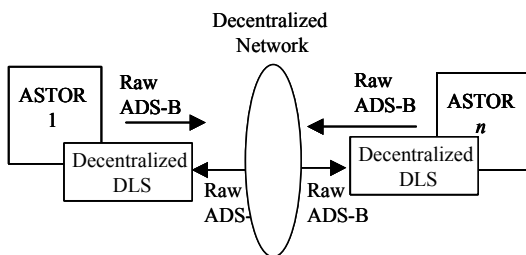


Figure 15. Illustration of a distributed DLS

creates a more complex software architecture since each aircraft now must incorporate a copy of the DLS model within its own process.

Network Load Comparison

To determine the network traffic of distributed and centralized systems, a series of experiments were performed between test processes designed to send and receive ADS-B messages. Two configurations were created. These were:

- 30 test aircraft and 500 background aircraft (generated with a target generator)
- 60 test aircraft and 500 background aircraft

An simplified data link representation was used (every aircraft sees every other aircraft) with an evenly distributed network usage (no spikes). In the first test, all messages were passed through a centralized processor. In the second test, messages were sent through a distributed network. The results are shown in Table 1. When using HLA in conjunction with Transmission Control Protocol (TCP), the centralized and distributed systems had nearly the same load with the centralized load being a bit more efficient. When User Datagram Protocol (UDP) is used however, the distributed system used an order of magnitude less bandwidth and required only a fractional increase to handle the 60 aircraft case over the 30 aircraft case. Therefore, maximum extensibility is achieved using the distributed design with a UDP transmission.

Table 1. Results from network load evaluation

Design and IPC	Usage (bits/sec)	
	30 aircraft	60 aircraft
Centralized with TCP	4.25E+7	8.73E+7
Distributed with TCP	4.80E+7	1.00E+8
Distributed with UDP	2.64E+6	2.74E+6

Distributed UDP Approach

From both a computational and network traffic point of view, the distributed solution using UDP is the most effective. Each ADS-B message is broadcast once (by each transmitting aircraft) and then received by every other aircraft federate. Then each federate is responsible for choosing its own subset of ADS-B messages. The advantage of the distributed UDP system is that the messages are each only broadcast once resulting in an $n \times 1$ solution. Furthermore, delays that might be incurred by double transmission of the same message are eliminated. However,

since message reception is not guaranteed, this approach may not guarantee the repeatability that is required. Therefore, while the UDP solution may provide the ultimate scalability performance, reliability is compromised.

Distributed TCP Approach

To maximize performance and reliability a distributed TCP approach still has the advantage of computational distribution while maintaining the message reliability of the TCP. Since repeatability of message transfer is critical to the success of the simulation, this approach is ultimately the one chosen.

Flight Information Service

Flight Information Services – Broadcast (FIS-B) is defined as the non-control information needed by pilots to operate in the National Airspace System (NAS) and internationally. Pilots, dispatchers, schedulers, and controllers all need accurate, timely FIS-B data to plan (or re-plan) and assess the execution of flight operations. Figure 16 illustrates the FIS-B system. FIS-B broadcasts weather information periodically through a VHF data link to the cockpit. FIS-B provides coverage over the continental US from 5000 ft. AGL to 17,500 MSL, except in areas unreachable due to mountainous terrain.

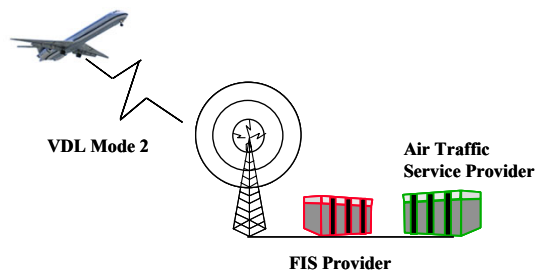


Figure 16. The FIS-B System

The actual FIS-B contains many assorted products. The ATOS FIS-B model contains a subset of these products that is relevant to the research to be conducted. Initially, the FIS-B products are the following:

- Hazardous Weather Advisory (SIGMETs)

- Special Use Airspace (SUA)
- Winds and Temperatures Aloft

Area Hazards

Both SIGMETs and SUAs are area hazards to be avoided by aircraft. To represent these advisories, a vector-based data format is used for the FIS-B area hazard message. The vector format allows for a series of connected line segments or a constraint polygon region to be identified using connected line segments described with a series of constraint vertices (defined by c_1, c_2, \dots, c_n). An illustration of a hazard is shown in Figure 17.

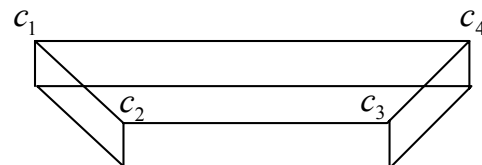


Figure 17. Example SIGMET polyhedrons modeled for ATOS.

Final Remarks

With the introduction of satellite navigation, ADS-B, and new CNS/ATM technologies, the airborne elements of the airspace system will have a much greater role in traffic management decision-making. As this role continues to increase in importance, ATOS will become increasingly more important in concept development and assessment, CNS requirements development, and the development of decision support technologies that support the system's human operators. ATOS Build 1 provides the foundation for an extensible and scalable multi-operator traffic operations simulation capability for investigation of distributed air/ground traffic management concepts of operation. Build 2 will introduce ground-based components and links to high-fidelity simulators, and Build 3 will provide links to research aircraft.

Acknowledgments

The ATOS was developed to support objectives of the NASA Advanced Air Transportation Technologies Project, a part of the Aviation

Systems Capacity Program. The Small Aircraft Transportation System Program and the Aviation Safety Program have also provided support for development of the simulation.

¹ Concept Definition for Distributed Air/Ground Traffic Management (DAG-TM), Version 1.0. Advanced Air Transportation Technologies Project, Aviation System Capacity Program, National Aeronautics and Space Administration, September 30, 1999.

² Ballin, M.G., Wing, D.J., Hughes, M.F. Conway, S.R.: Airborne Separation Assurance and Traffic Management: Research of Concepts and Technology, AIAA-99-3989, 1999.

³ Weske, R.A., and G.L. Danek,: Pseudo Aircraft Systems: A Multi-Aircraft Simulation System for Air Traffic Control Research, Proceedings of the AIAA Flight Simulation Technologies Conference, Monterey, CA, August 1993.

⁴ Target Generation Facility (TGF) Overview, Aviation Simulation and Human Factors Division, FAA William J. Hughes Technical Center, Atlantic City, NJ, September 1996.

⁵Peters, M. E., The Engineering Analysis and Design of the Aircraft Dynamics Model For the FAA Target Generation Facility. Tech. Report 99162-01, Seagull Technology, Los Gatos, CA, August 1999.

⁶ Miles, E. S., Davis, P. D., and Wing, D. J.: Development of a Free-Flight Simulation Infrastructure. AIAA-99-4193, July 1999.

⁷ Mondoloni, S., Palmer, M.T., and Wing, D. J.: Development of a Prototype Airborne Conflict Detection and Resolution Capability. AIAA-2002-4446, August 2002.

⁸ Wing, D. J., Adams, R. J., Barmore, B. E., and Moses D.: Airborne Use of Traffic Intent Information in a Distributed Air/Ground Traffic Management Concept: Experiment Design and Preliminary Results. 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe, December 2001.

⁹ RTI 1.3-Next Generation Programmer's Guide, Version 4. Defense Modeling and Simulation Office, Alexandria, VA, June, 2001.

¹⁰ Guidance for use of Avionics Equipment and Software in Simulators. ARINC Report 610A, February 1, 1994.

¹¹ Leslie, R.; Geyer, D.; Cunningham, K.; Madden, M.; Kenney, P.; and Glaab, P.: LaSRS++: An Object-Oriented Framework for Real-Time Simulation of Aircraft, AIAA-98-4529, Modeling and Simulation Technology Conference, Boston, MA, August, 1998.

¹² Ballin, M.G., Sharma, V., Vivona, R., Johnson, E.J., Ramiscal, E., A Flight Deck Decision Support Tool For Autonomous Airborne Operations, AIAA-02, July, 2002.

¹³Contract Task Order 4, Preliminary Design Review. Advanced Air Transportation Technologies /Air Traffic Management System Development & Integration, Contract No. NAS2-00015, NASA Langley Research Center, Hampton, VA, November, 2001.